Глибока біометрична автентифікація та безпека

Distributed Lab

14 березня 2024 р.

План

Вступ

Формулювання задачі

Зазвичай, на вхід подається набір даних виду:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\},$$

де задача – побудувати функцію f, що "достатньо точно" відображає x_i на y_i (*Supervised Learning*).

Приклад

Розпізнавання цифри з зображення. x_i (вхід) – зображення, y_i (вихід) – цифра від 0 до 9.

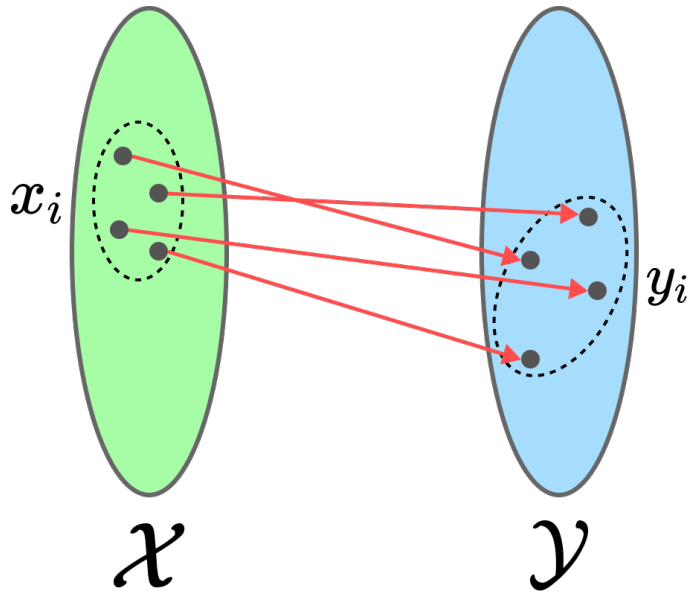


Рис.: Вхідний набір данних $\mathcal{D}$ – послідовність пар виду $x_i \mapsto y_i$.

Приклад: MNIST

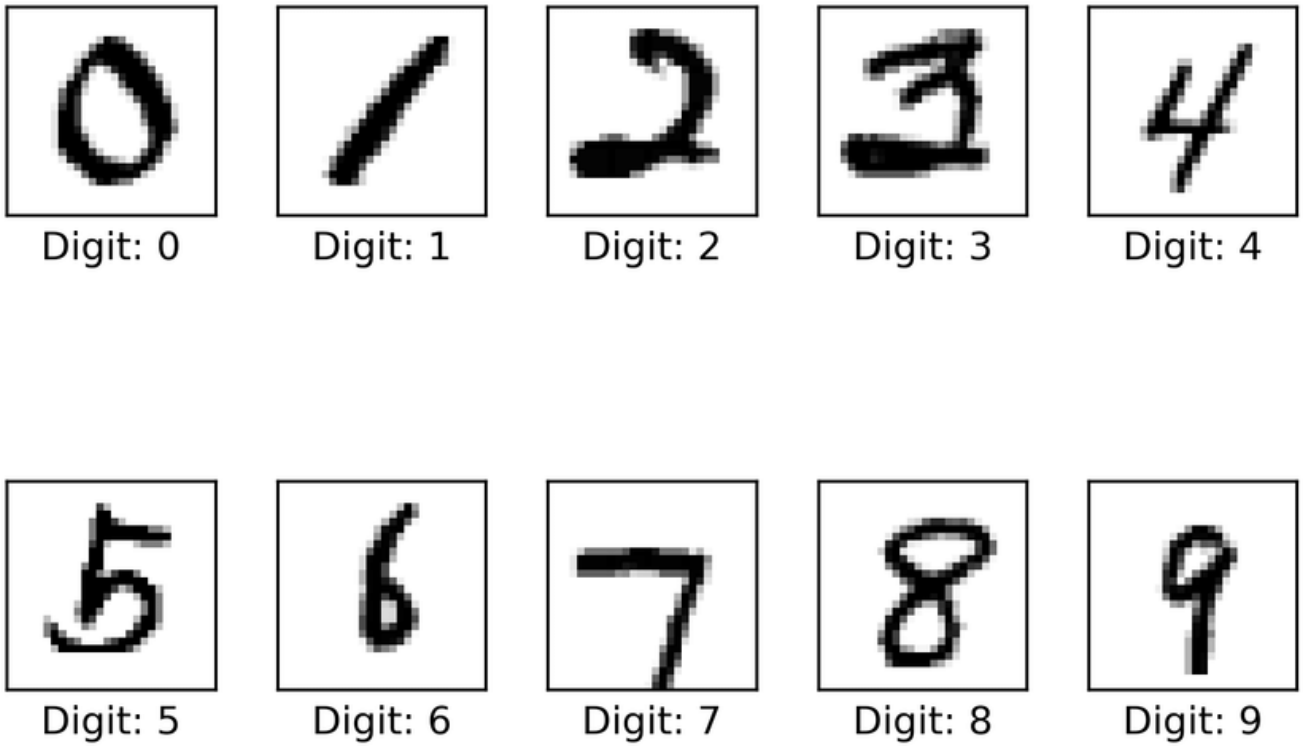


Рис.: Розпізнавання цифр. Набір данних MNIST – найбільш популярний вибір для написання прототипів (*Proof of Concept*).

Ефективність відображення

Питання

Як оцінити, що задана функція f є “гарною” або “поганою”?

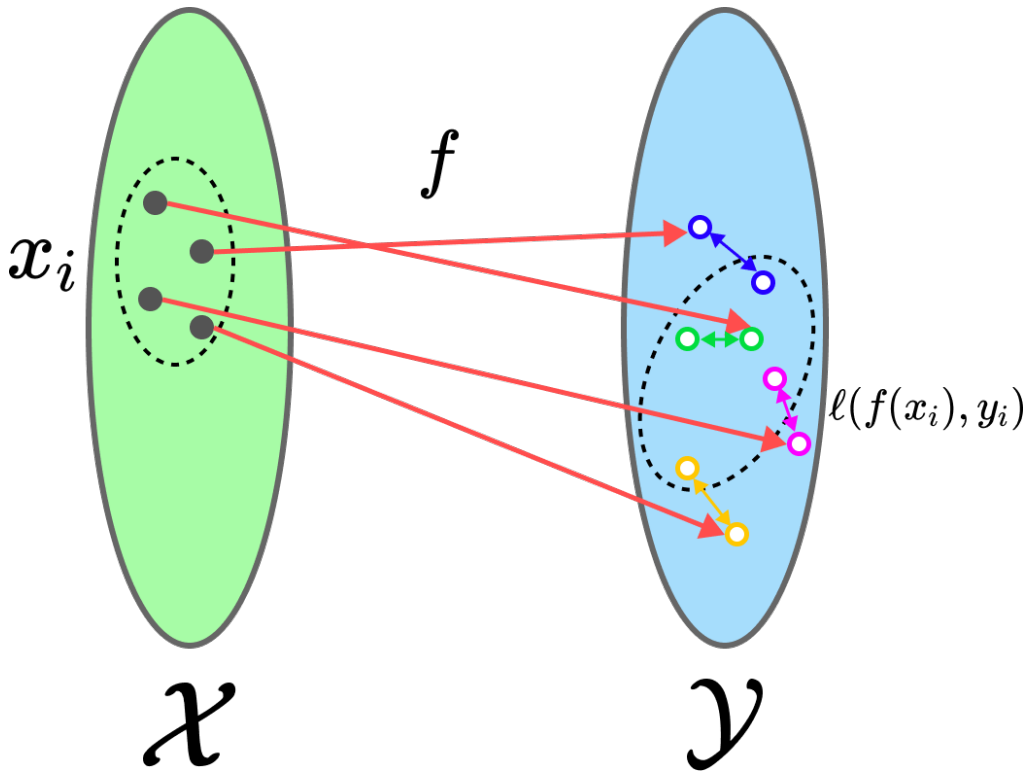


Рис.: $f(x_i)$ “промахнулось” і дало відхилення від y_i – що далі?

Функція втрати

✓ Введемо втрату ℓ – міра відстані між передбаченням $\hat{\mathbf{y}} = f(\mathbf{x})$ та фактичним значенням $\mathbf{y}$.

Приклад функції втрати

Якщо вихідне значення – вектор, то можна покласти $\ell(\mathbf{y}, \hat{\mathbf{y}}) := \text{відстань}(\mathbf{y}, \hat{\mathbf{y}})$.

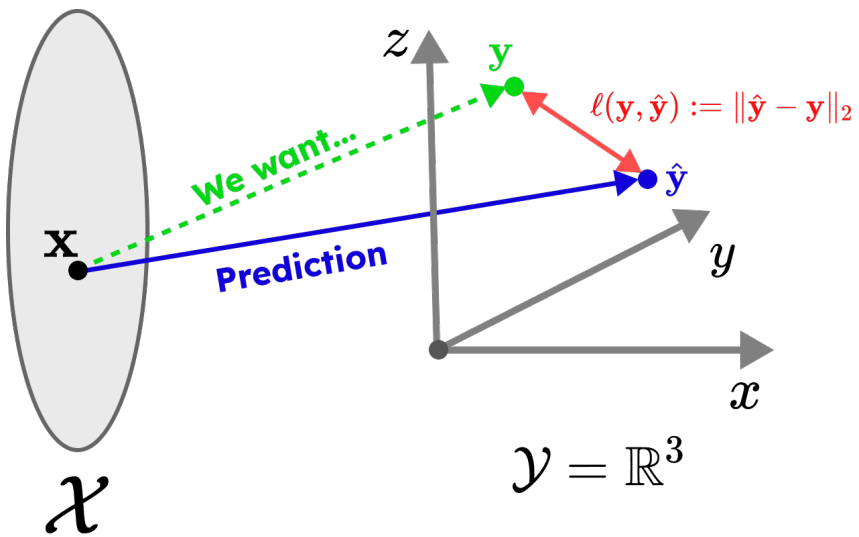


Рис.: Для $\mathcal{Y} = \mathbb{R}^3$ беремо Евклідову відстань для ℓ.

Оптимізаційна задача

Нехай функція $f(\star;\theta)$ параметризована набором параметрів θ.

Приклад параметризації

Наприклад, нехай ми шукаємо

$$f(x) = ax + b,$$

де $\theta = (a, b)$

Задача – мінімізувати втрату $\ell(f(x_i), y_i)$.

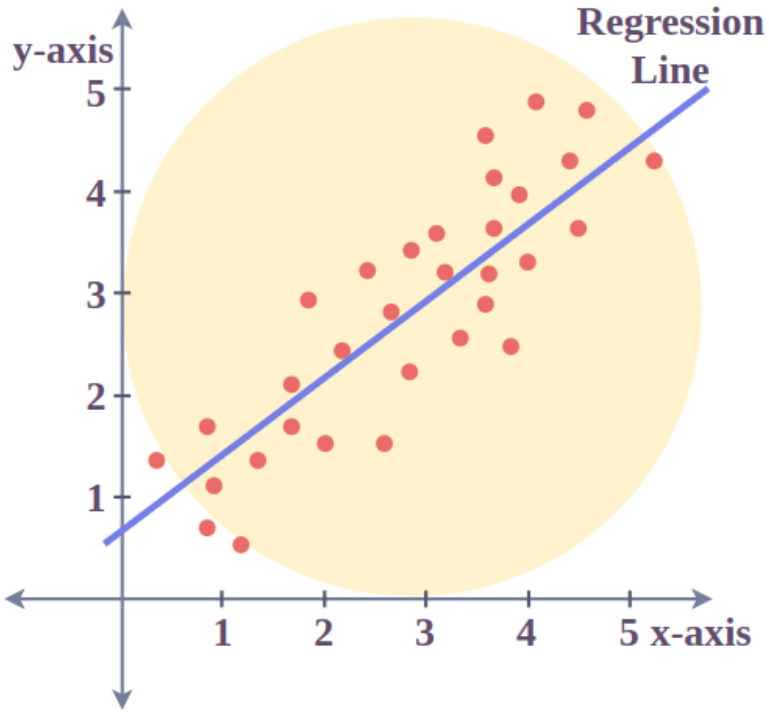


Рис.: Підбираємо пряму, що проходить максимально близько до інших точок.

Повнозв'язні нейронні мережі (Dense Neural Networks)

Будуємо функцію, що на вхід приймає вектор довжини n_{in}, на вихід видає вектор довжини n_{out}, а також параметризована матрицями ваг та зсувами (bias).

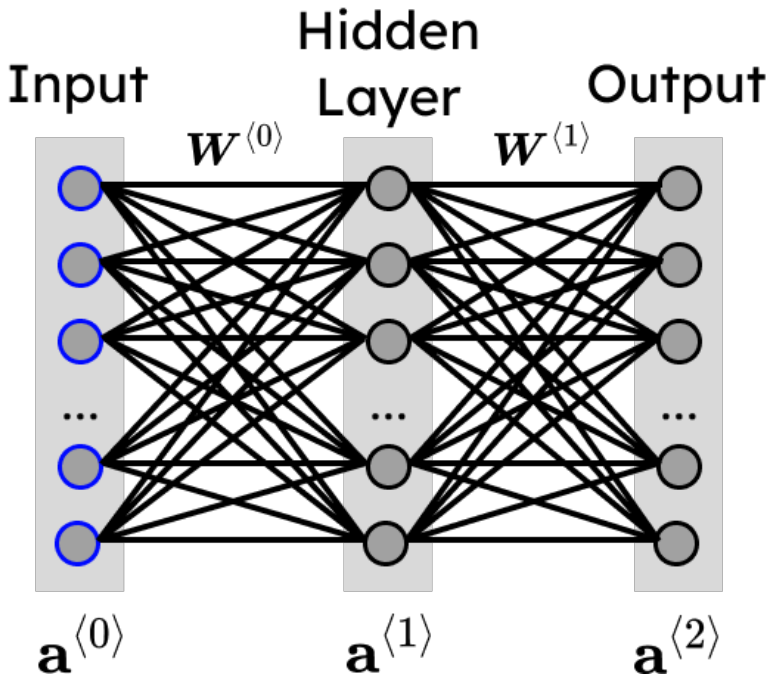


Рис.: Приклад повнозв'язаної нейронної мережі з трьома шарами.

Вхідні нейрони

Що таке нейрон?

Нейрон – структурна одиниця нейронної мережі. По своїй суті – вершина, що містить число – активацію.

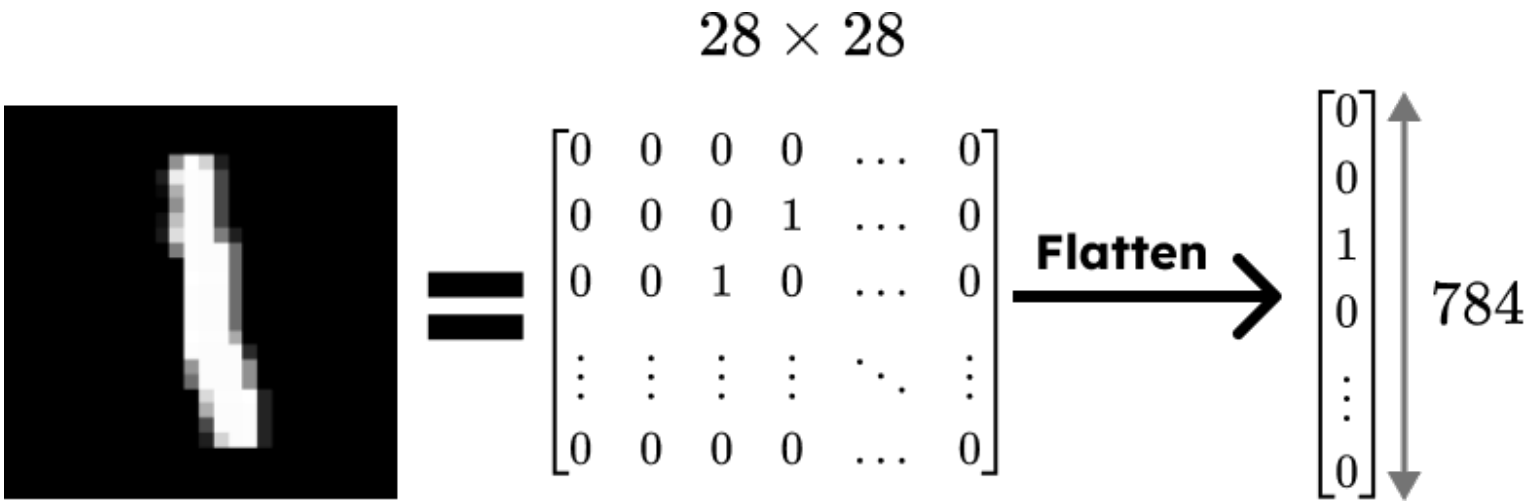


Рис.: Нейрони у першому шарі. Матрицю зображення перетворюємо у плоский вектор.

Forward Propagation

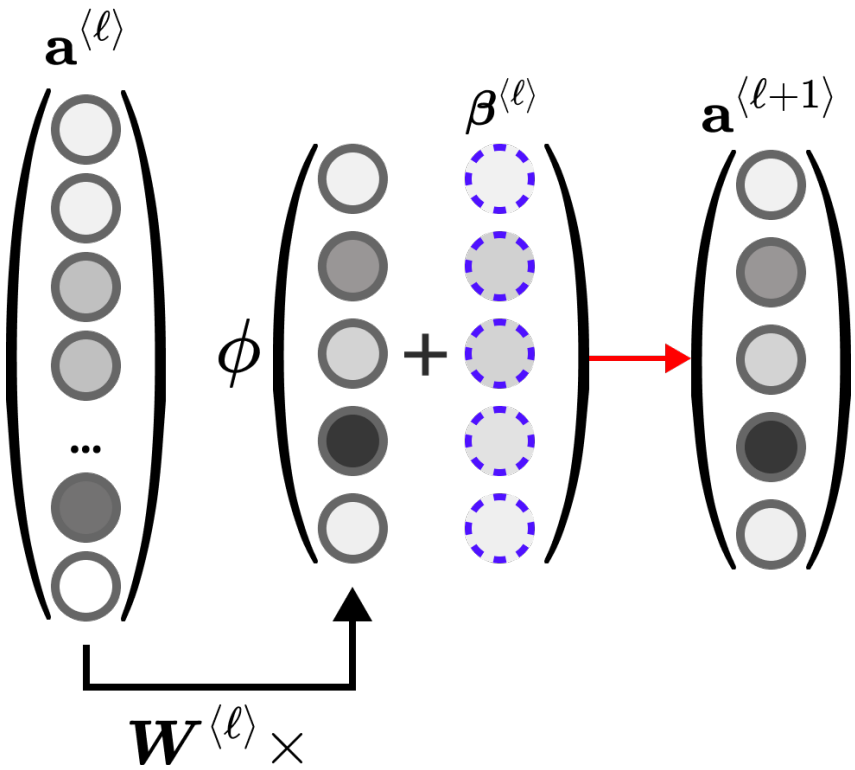


Рис.: Пряме поширення. Спочатку знаходимо матричний добуток, додаємо зсув і накладаємо нелінійну функцію.

Пряме поширення: обрахунок втрати

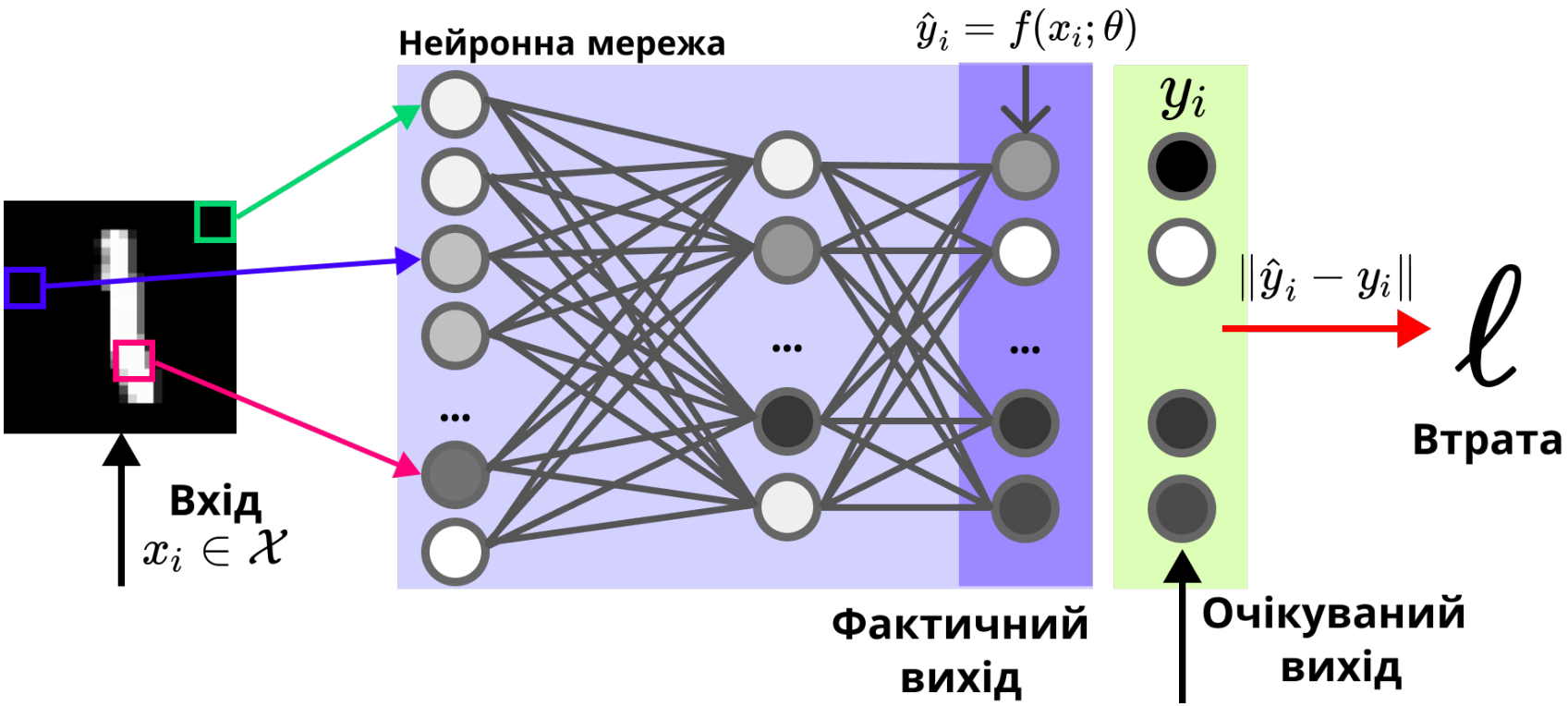


Рис.: Пряме поширення: обрахунок значення втрати

Резюме

- Нейронна мережа f – багатопараметризована функція.
- Dense нейронна мережа приймає вектор і “випльовує” також вектор.
- Маючи набір данних, ми намагаємося мінімізувати певну задану функцію ℓ, підбираючи параметри у сімействі функцій.

Мотивація використання конволюційної нейронної мережі

630

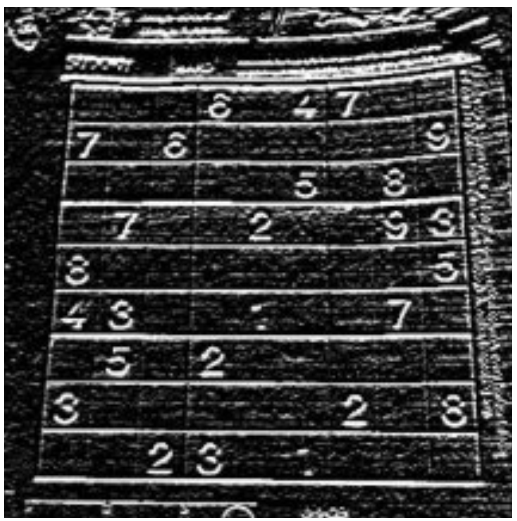
360

630 × 360 = 226,800

Фільтри Собеля

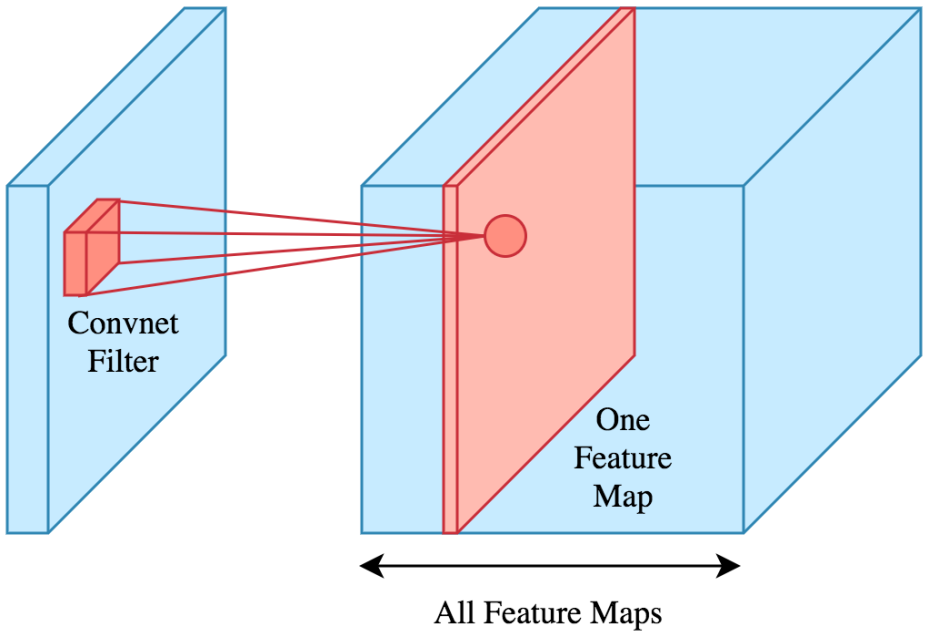
$$* \underbrace{\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}}_{x \text{ Sobel kernel}} =$$

$$* \underbrace{\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}}_{y \text{ Sobel kernel}} =$$



Конволюційний шар

- Фільтр по зображенню $W \times H \times n_C$ з n_C каналами – це набір з n_C фільтрів.
- Один набір – один шар у вихідному "об'ємі". n_f фільтрів дає вихід з n_f каналами.
- Тренувальні параметри – параметри фільтрів, зсуви (один на кожен фільтр), гіперпараметр – активаційна функція.



Max Pooling

1. Рухаємо $2 \times 2 \times n_C$ фільтр по зображенню.
2. Взяти максимальний елемент на кожному каналі і записати у вихідне зображення.

Навіщо це треба? Ми зменьшуємо зображення в 4 рази.

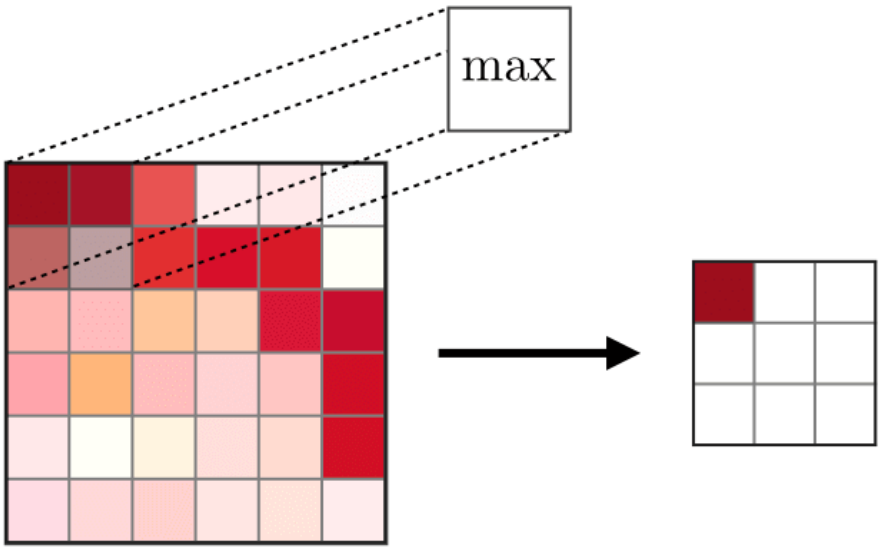


Рис.: Ілюстрація MaxPool шару. Зображення взято з Afshine Amidi, Stanford CS 230 – Deep Learning, CNN Cheatsheat

Конволюційна нейронна мережа, підсумовуючи

1. Використовуємо конволюційні шари (Conv2D).
2. Зменьшуємо розмір зображення за допомогою `Conv2D` з $s = 2$ або `MaxPool` шаром.
3. Повторити, поки об'єм зображення не стане достатньо маленьким.
4. Перевести у повнозв'язану нейронну мережу $\Longrightarrow$ вихід.

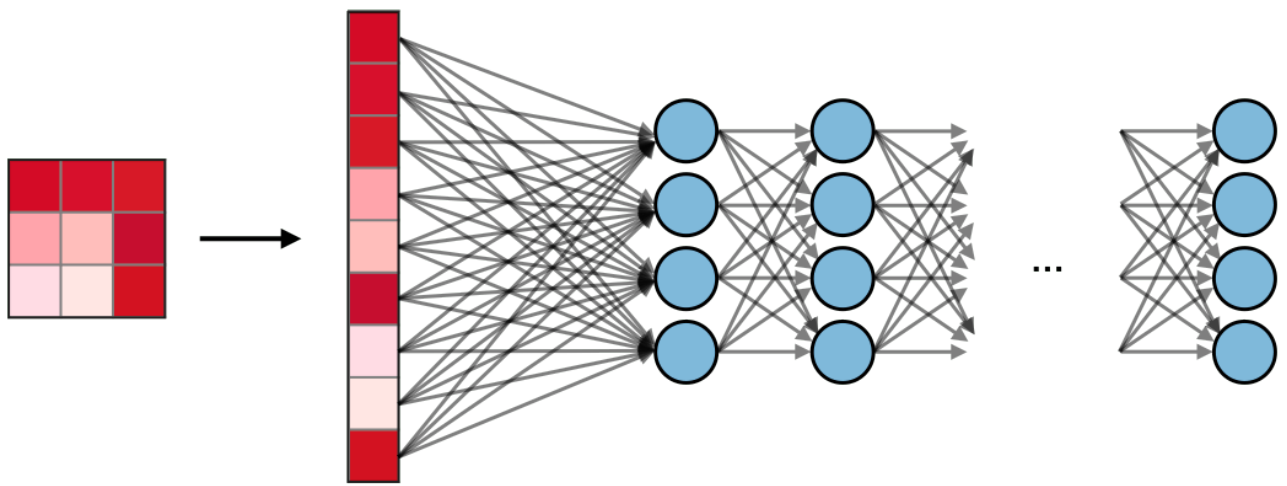


Рис.: Повнозв'язана нейронна мережа в кінці CNN. Зображення взято з Afshine Amidi, Stanford CS 230 – Deep Learning, CNN Cheatsheat

Конволюційна нейронна мережа, підсумовуючи

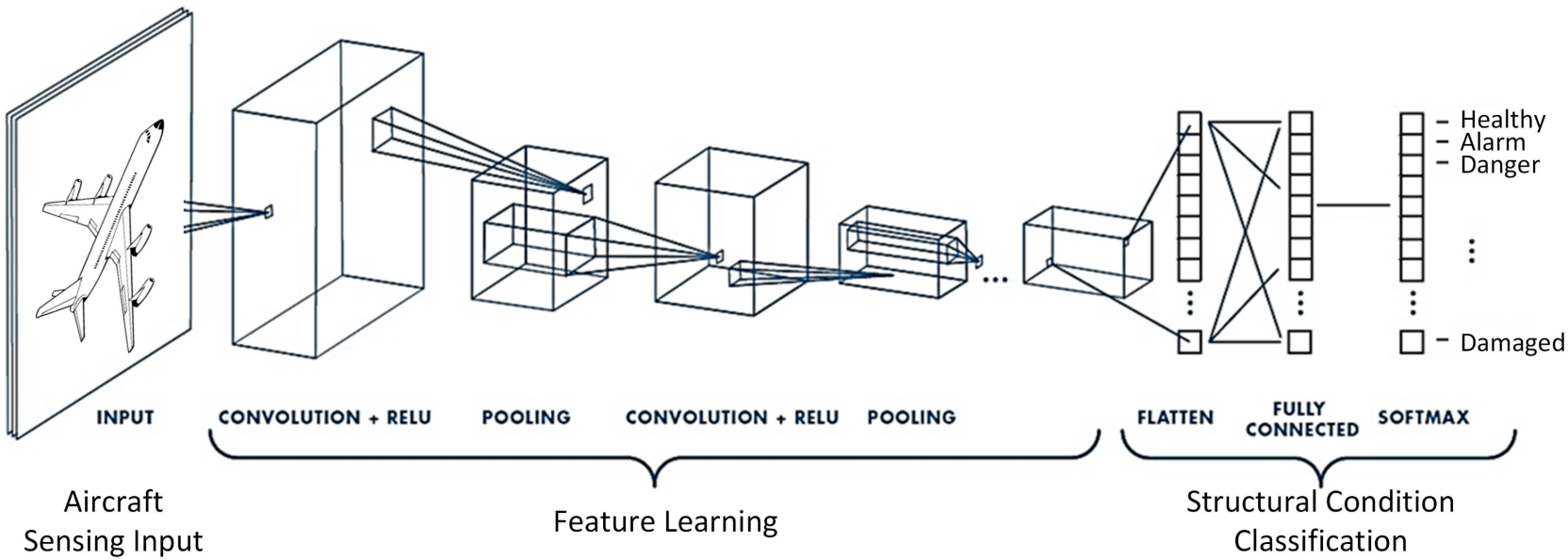
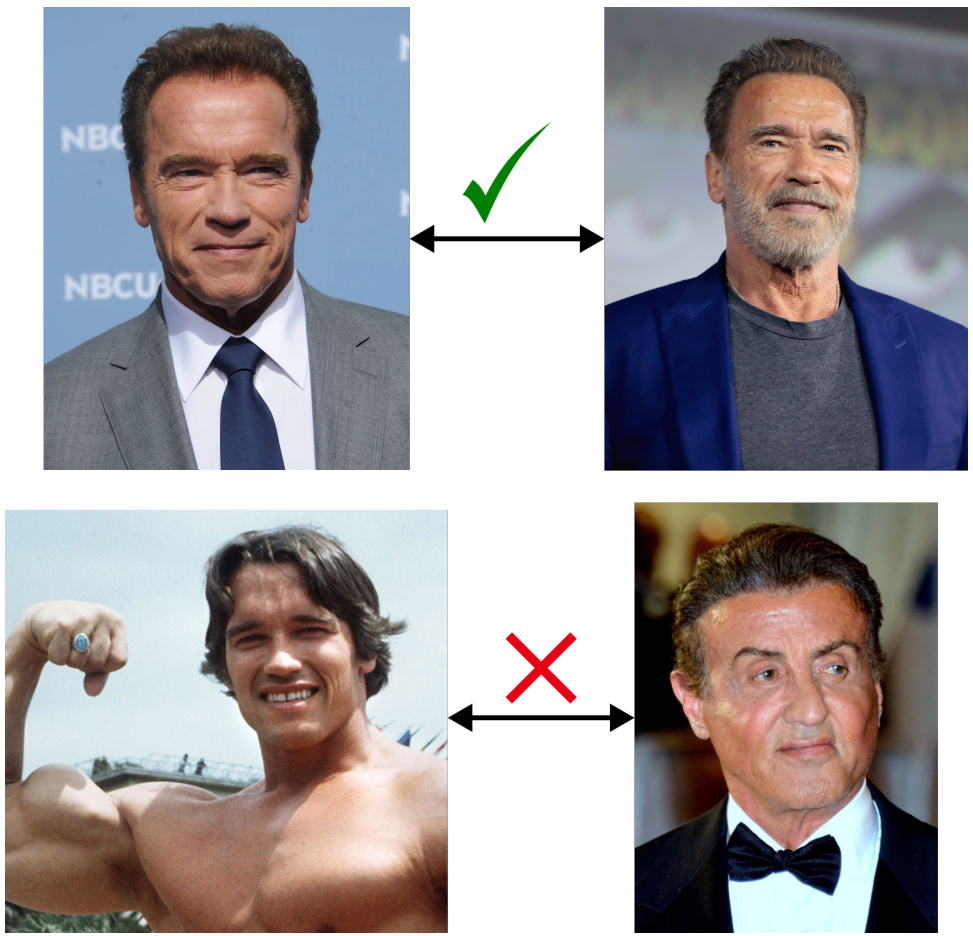


Рис.: Повний вигляд конволюційної мережі. Зображення взято з роботи Iuliana Tabian et al. 2019. A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures.

Deep Pattern Recognition

Постановка задачі

Маючи два зображення X, Y, сказати чи відповідають вони одній людині чи ні.



Перша ідея

Нехай кожній людині відповідає номер. Тоді, будуємо класифікаційну нейронну мережу (C – кількість класів):

$$\mathcal{F} : \text{Image} \to \{1, \ldots, C\}$$

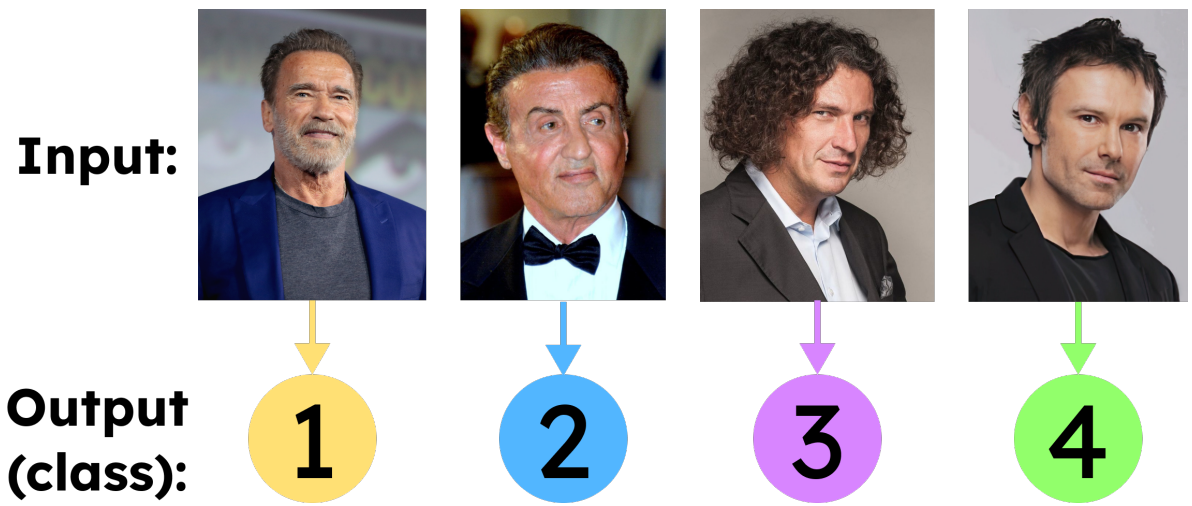


Рис.: Даємо номер кожній людині і будуємо мультикласифікаційну модель.

Чому ні?

- Людей м'яко кажучи багато – близько 8.1 млрд на момент написання цієї презентації :)
- Навіть маючи фіксований набір людей, потрібно більше 50-100 фотографій на людину (One-shot problem).

Рис.: Набір класів не є фіксованим, інакше нам доведеться зафіксувати 8+ млрд класів

Друга ідея: Embedding Neural Network

Будуємо $\mathcal{F} : \text{Image} \to S^{m-1}$. Вперше запропоновано в Florian Schroff et al. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. 2015.

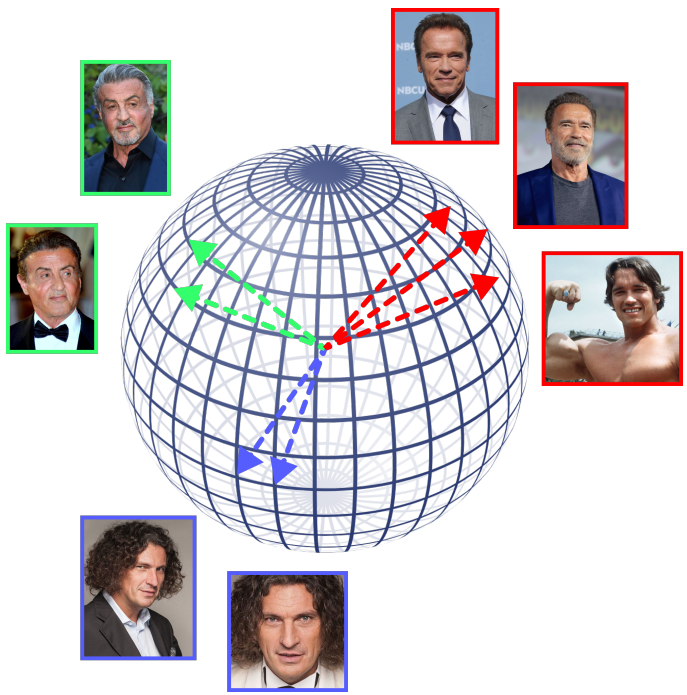


Рис.: Вихід функції (вектор фіч або “embedding vector”) буде давати “характеристику” людини.

Embedding Neural Network: інтуїція

Приклад вектору фіч – це набір відстаней між ключовими точками.

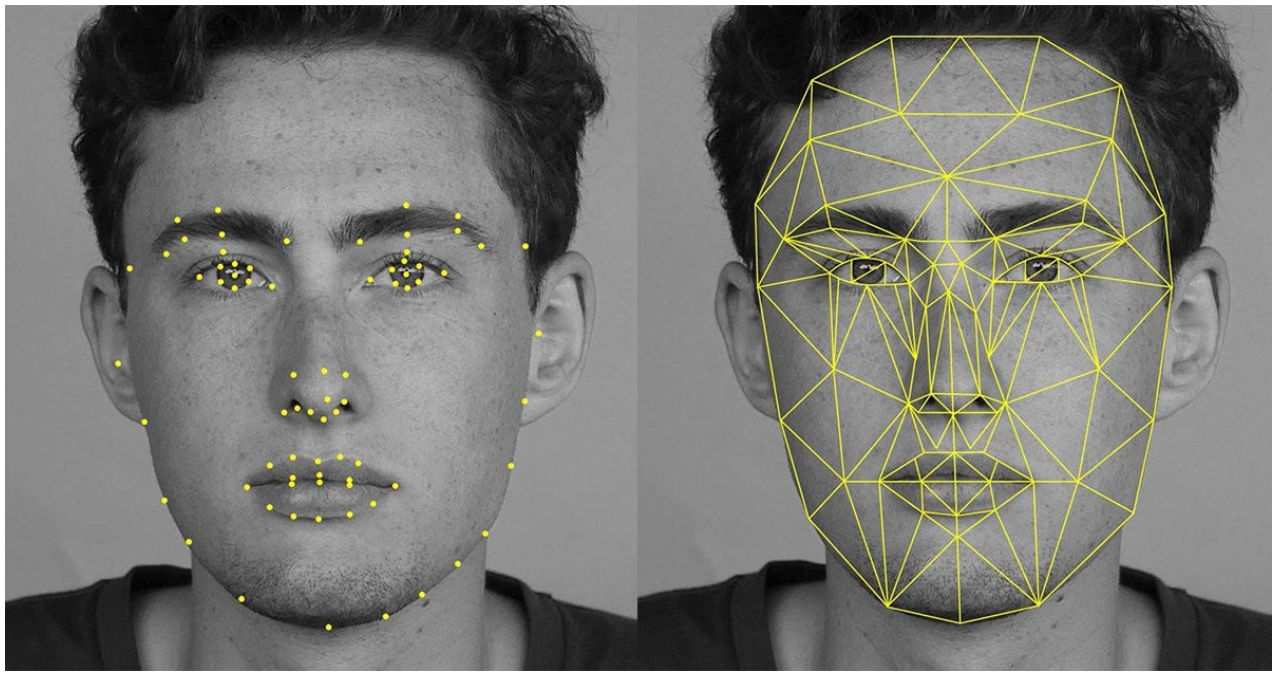


Рис.: Ключові точки на обличчі.

Ілюстрація роботи Embedding нейронної мережі

Приклад

Нехай на вхід ми отримали зображення X, Y, Z і для $m = 3$ ми отримали наступні вектори фіч:

$$\mathbf{x} = \langle 0.568, 0.568, 0.596 \rangle$$
$$\mathbf{y} = \langle 0.613, 0.529, 0.585 \rangle$$
$$\mathbf{z} = \langle -0.408, -0.816, 0.408 \rangle$$

Які два зображення з набору X, Y, Z належать одній людині, а яка одна до іншої?

Добре видно, що X, Y належать одній людині. Але як ми це визначили?

Метрика схожості

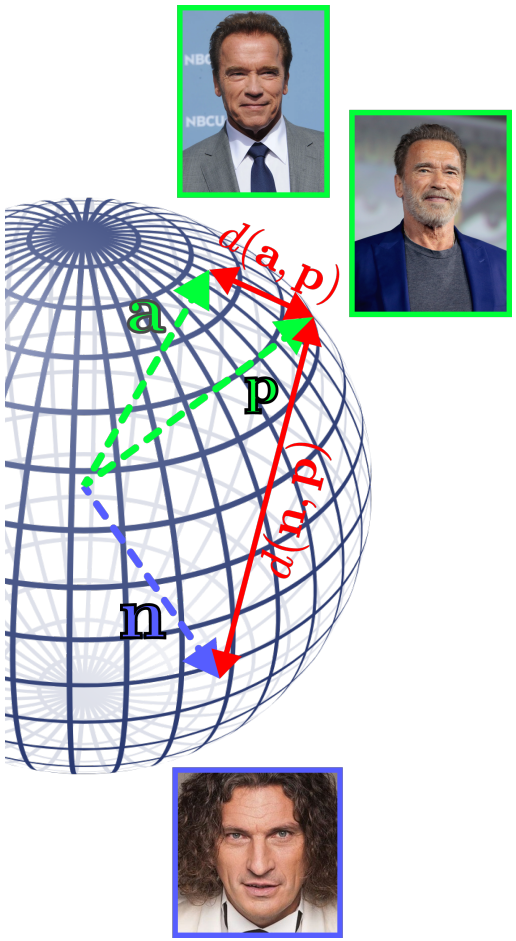


Рис.: Метрика різниці людей – відстань між векторами фіч

Введемо відстань між векторами фіч $d(\cdot, \star)$. Найбільш частий вибор – Евклідова метрика.
Умова на одну людину:

$$\boxed{d(\mathcal{F}(X), \mathcal{F}(Y)) \leq \tau},$$

де τ – так званий "поріг" або threshold.

Як реалізувати? Псевдокод

Реєстрація:

1. Прочитати зображення людини X зі сканеру.
2. Додати до бази данних $\mathcal{F}(X)$.

Автентифікація:

1. Прочитати зображення людини Y зі сканеру.
2. Знайти вектор $y = \mathcal{F}(Y) = (y_1, \ldots, y_{128})$.
3. Для кожного вектору $z = (z_1, \ldots, z_{128})$ з бази даних зробити наступну дію:
   1. Якщо $\sum_{i=1}^{128}(y_i - z_i)^2 < \tau$ – впустити людину.
   2. Якщо ні, то продовжити.

Резюме

Всі ми – просто набір 128 дійсних чисел на гіперсфері :(

Як навчати? Головна ідея

Візьмемо три фотографії:

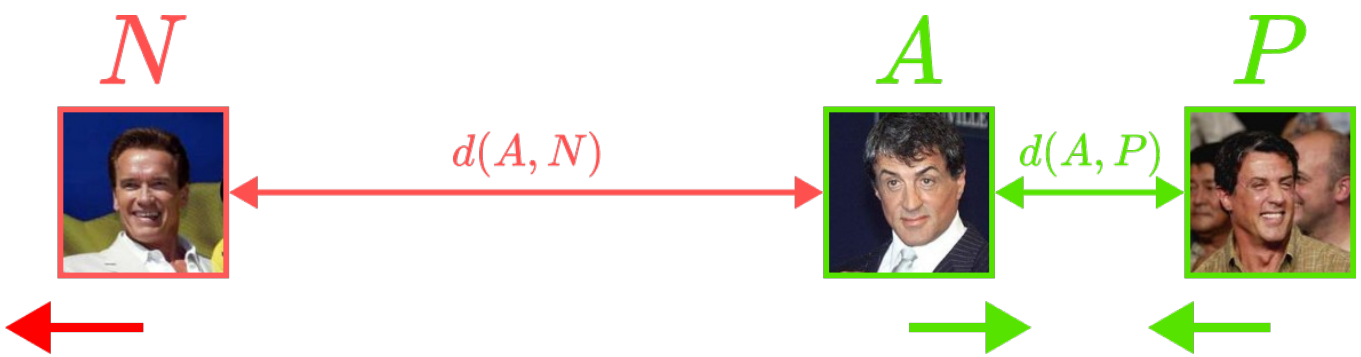
- A (anchor) – зображення людини 1.
- P (positive) – інше зображення людини 1.
- N (negative) – зображення людини 2.

Головне, що ми хочемо:

$$d(A, P) < d(A, N)$$

Або, більш строга умова умова:

$$d(A, P) < d(A, N) - \gamma$$



Як навчати? Робимо метрику!

Як з цієї ідеї зробити конкретну метрику (втрату)?
Використовується наступна функція:

$$\ell(A, P, N) = \max\{d(A, P) - d(A, N) + \gamma, 0\}$$

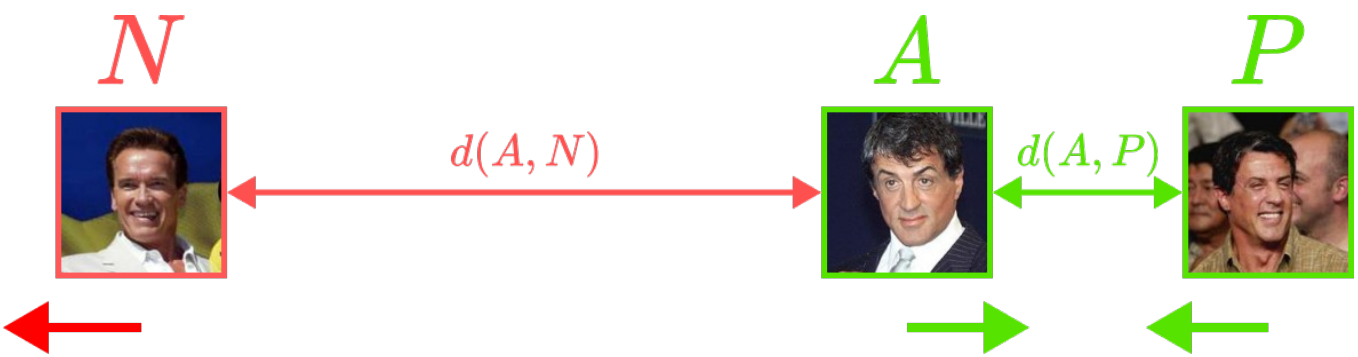


Рис.: Після застосування градієнту, ми хочемо віддалити (A, N) і наблизити (A, P).

Як навчати? Triplet Network

Як побудувати нейронну мережу? Triplet Network!

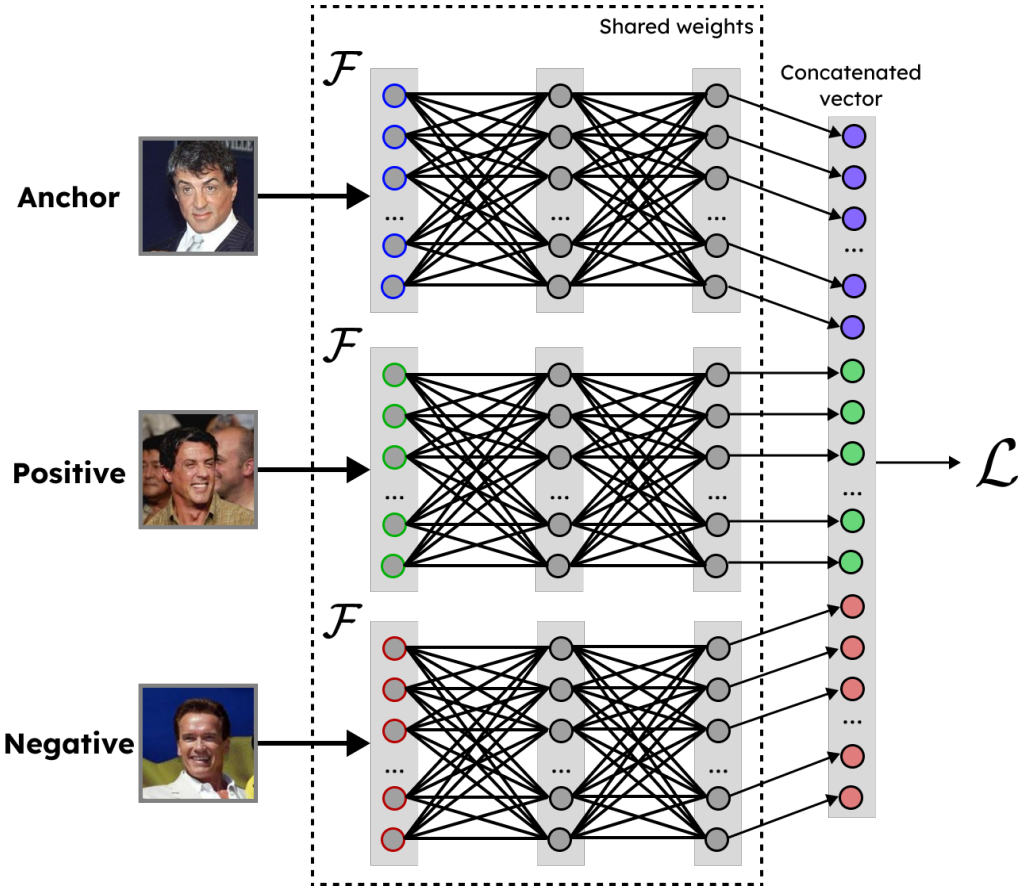


Рис.: Triplet Network архітектура.

Ідентифікуємий, але не впізнаємий

Безпека векторів фіч

- Зберігати зображення в базі данних небезпечно.
- Чи безпечно зберігати вектори фіч? Ні.
- Чи можна створити криптографічний примітив? Дуже активна робота саме в цьому напрямку!

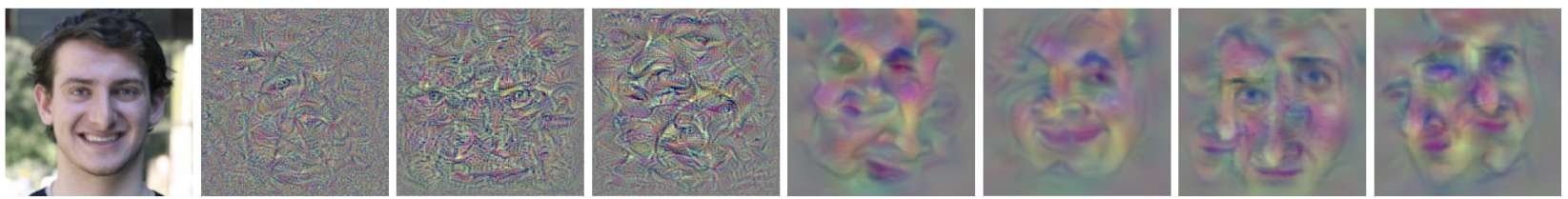


Figure 2: (a) is the desired image whose FaceNet embedding is used. (b) and (c) are generated using just facenet embedding loss. (d), (e), (f) are geenerated using increasing total variation loss, making an increasingly smother and more image. (g) and (h) are generated using guiding image loss on an intermediate FaceNet layer. We observe increasing identifiability, but not a convincing result.

Рис.: Знаходження обличчя з векторів фіч. Взято з Edward Vendrow et al. 2018. Inverting Facial Embeddings with GANs

Безпека векторів фіч

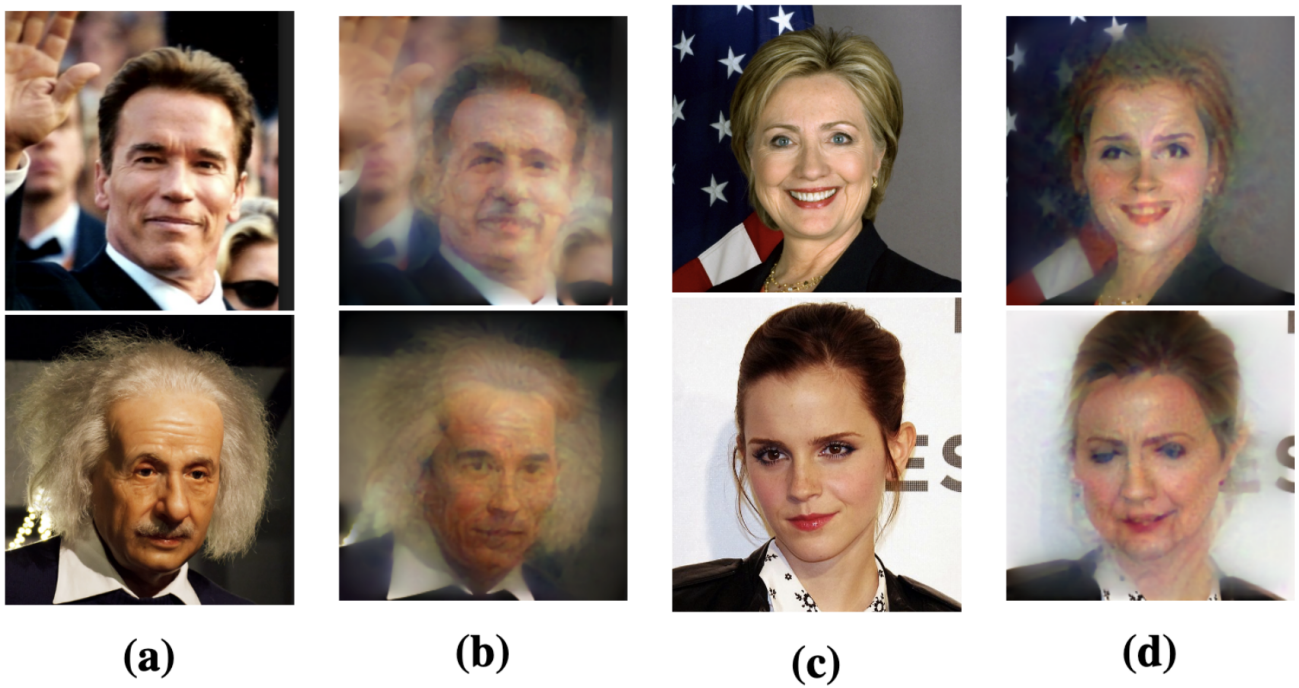


Рис.: Обернення FaceNet, використовуючи базове зображення. Andrey Zhmoginov et al. “Inverting face embeddings with convolutional neural networks”

Відміняєма біометрія

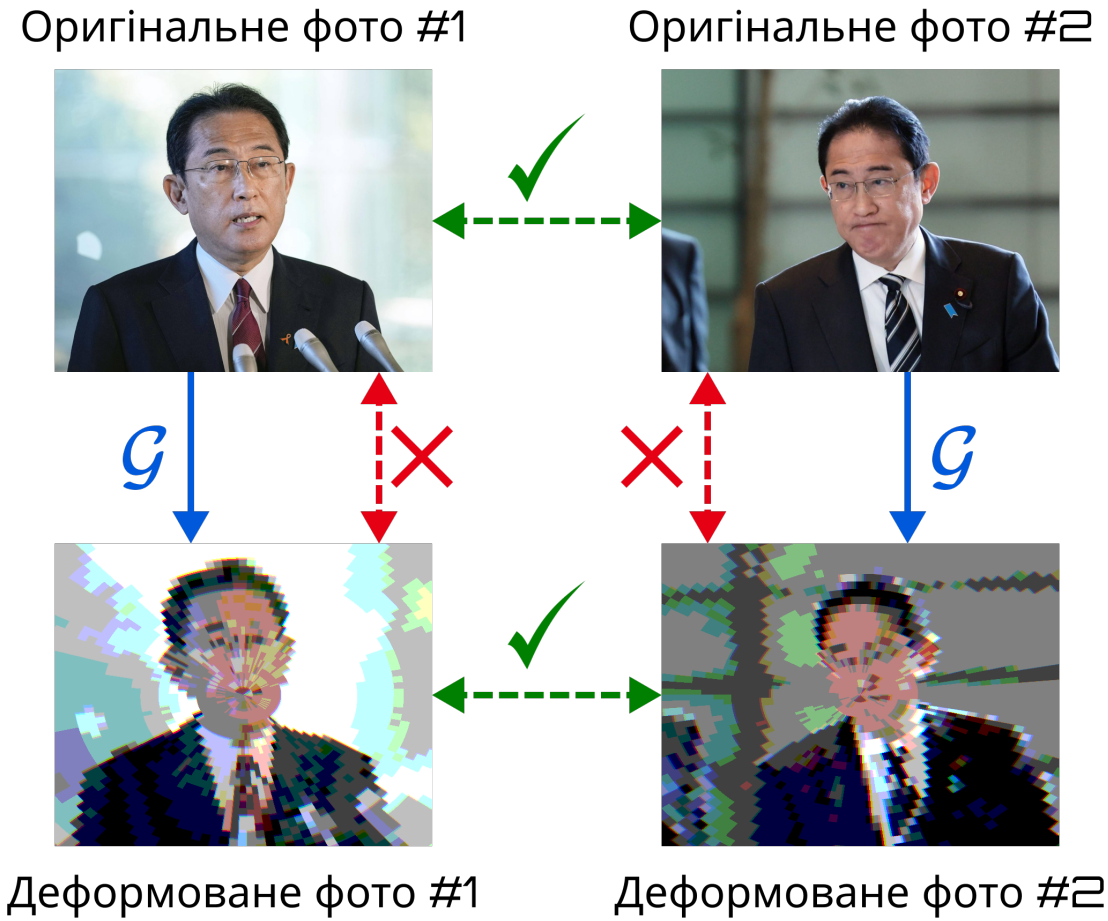


Рис.: Система відміняємої біометрії. Функція $\mathcal{G}$, переводить зображення у простір деформованих фото, в якому можна порівнювати зображення.

Шифрування

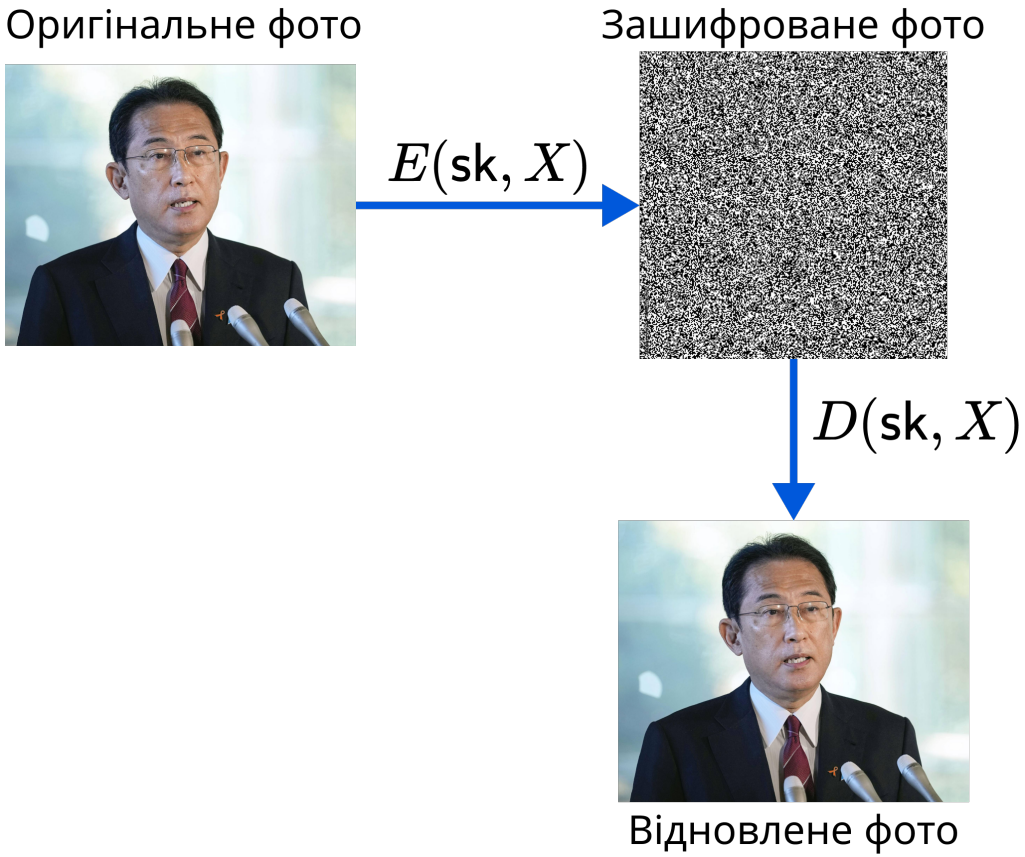


Рис.: Система (симетричного) шифрування. Маємо функцію шифрування $E(\mathsf{sk}, X)$ та дешифрування $D(\mathsf{sk}, C)$ з умовою коректності $D(\mathsf{sk}, E(\mathsf{sk}, X)) = X$.

Проблема

Нехай ми подали зображення X на вхід і хочемо порівняти з зашифрованним шаблоном T в базі данних.

- *Відміняєма біометрія* спочатку трансформує зображення $\mathcal{G}(X)$, а потім порівнює з шаблоном: $d(\mathcal{G}(X), T) \overset{?}{\leq} \tau$.
- *Шифрування* спочатку декодує зображення $D(\mathrm{sk}, T)$, а потім порівнює з вхідним: $d(D(\mathrm{sk}, T), X) \overset{?}{\leq} \tau$.

В будь-якому випадку, маємо дві дії: предобробка зображення, а потім порівняння. Чи можна це звести у лише одне порівняння: $d^*(X, T) \overset{?}{\leq} \tau$?

Наше рішення

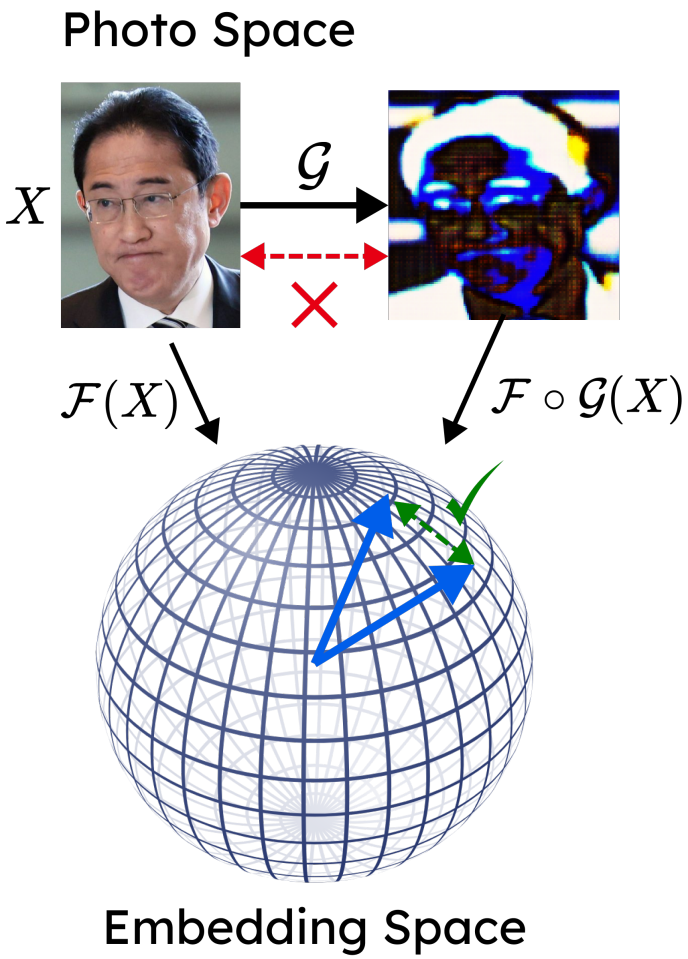


Рис.: Ілюстрація нашого методу.

Система захисту складається з пари $(\mathcal{G}, \mathcal{F})$, де:

- $\mathcal{G}$ – генератор деформованих фото;
- $\mathcal{F}$ – embedding нейронна мережа.

Головні властивості:

- $\mathcal{F}(X) \approx \mathcal{F}(\mathcal{G}(X))$.
- $\mathcal{G}^{-1}$ "складно" обрахувати.

Оптимізаційна задача – візуалізація

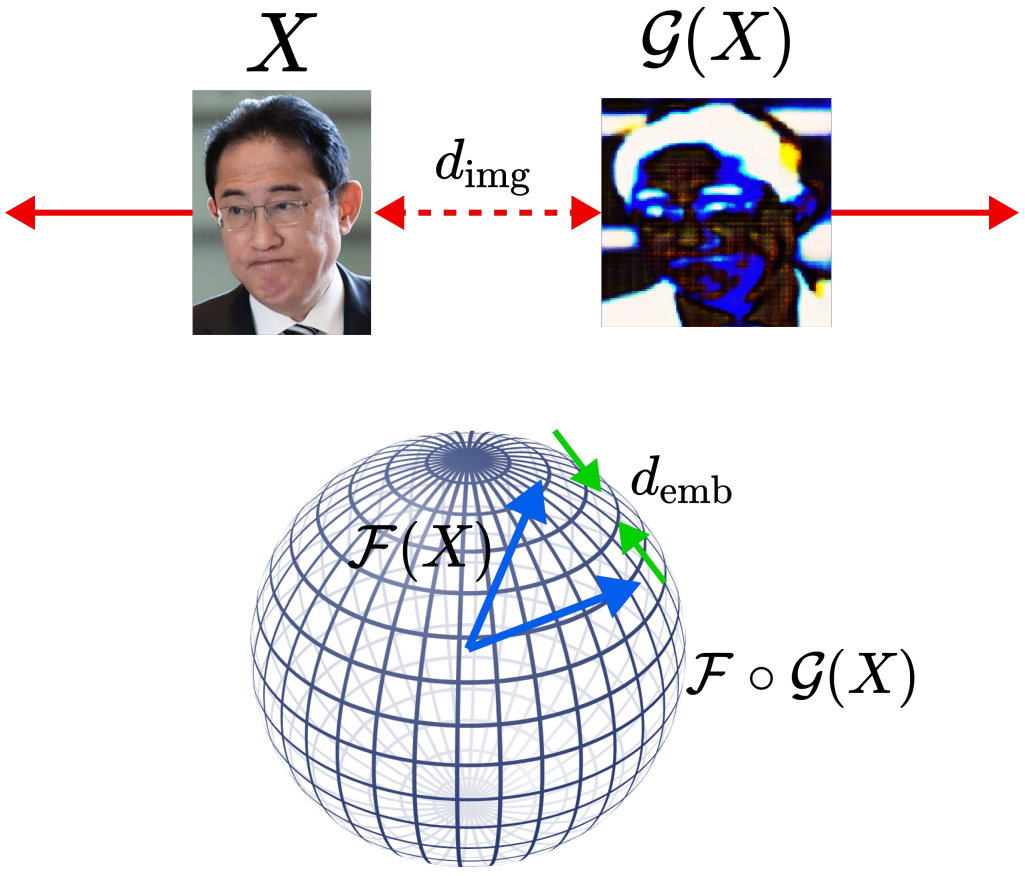


Рис.: Візуалізація цілі задачі.

Trainer Network

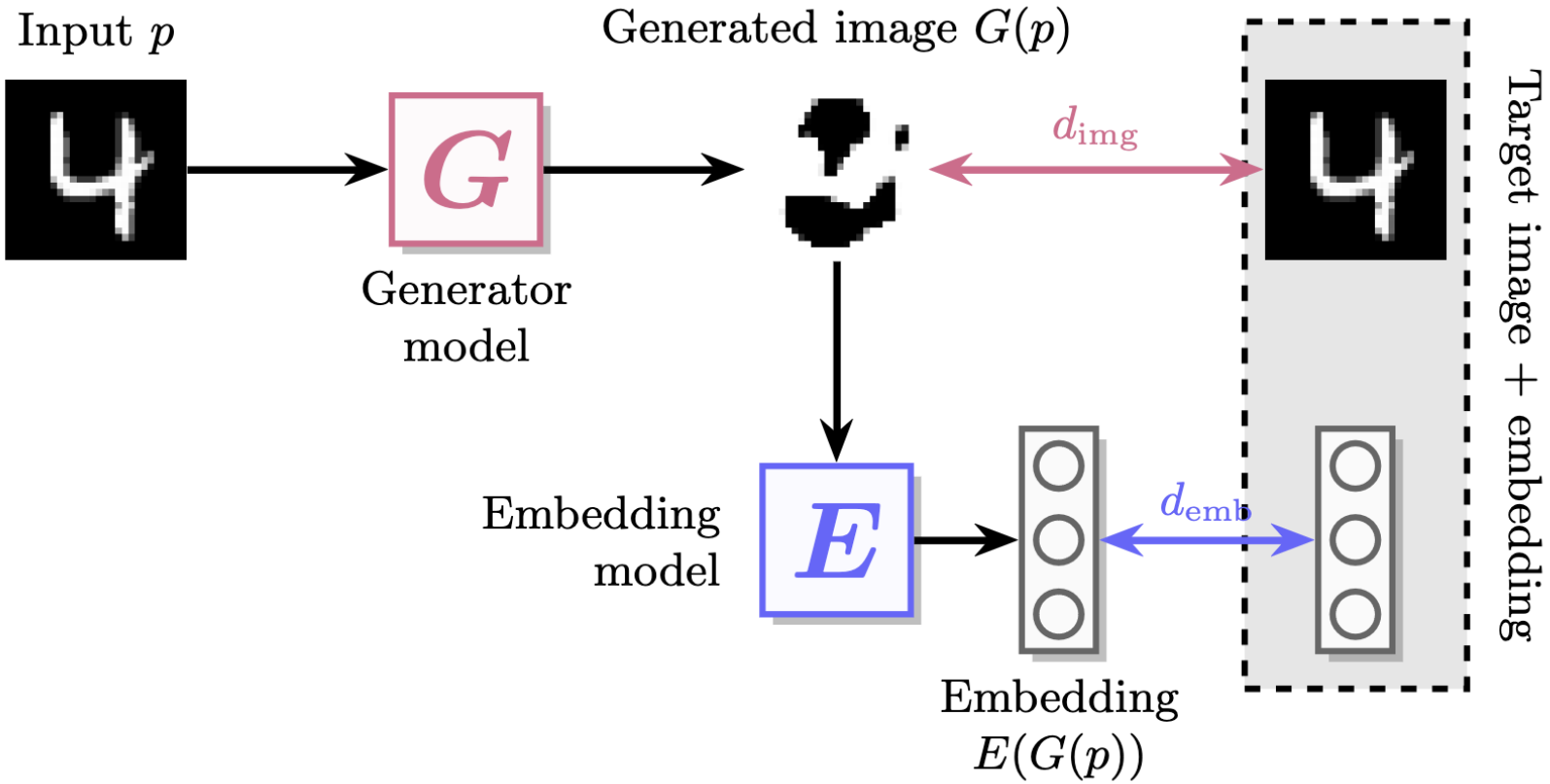


Fig. 2: Trainer Network architecture

Рис.: Ілюстрація Trainer Network архітектури з нашої роботи.

U-Net

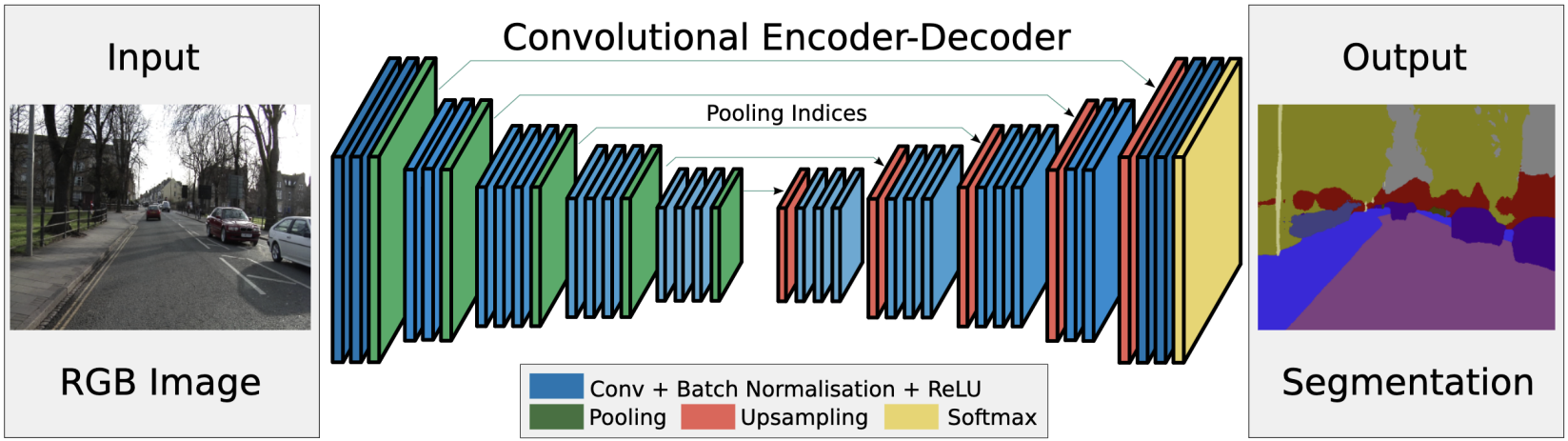


Рис.: U-Net архітектура (Encoder → Decoder). Спочатку зображення стискається, а потім розтискається. Ілюстрація взята з K. Murphy "Probabilistic Machine Learning: An introduction". MIT Press. 2022

Приклади зображень. MNIST

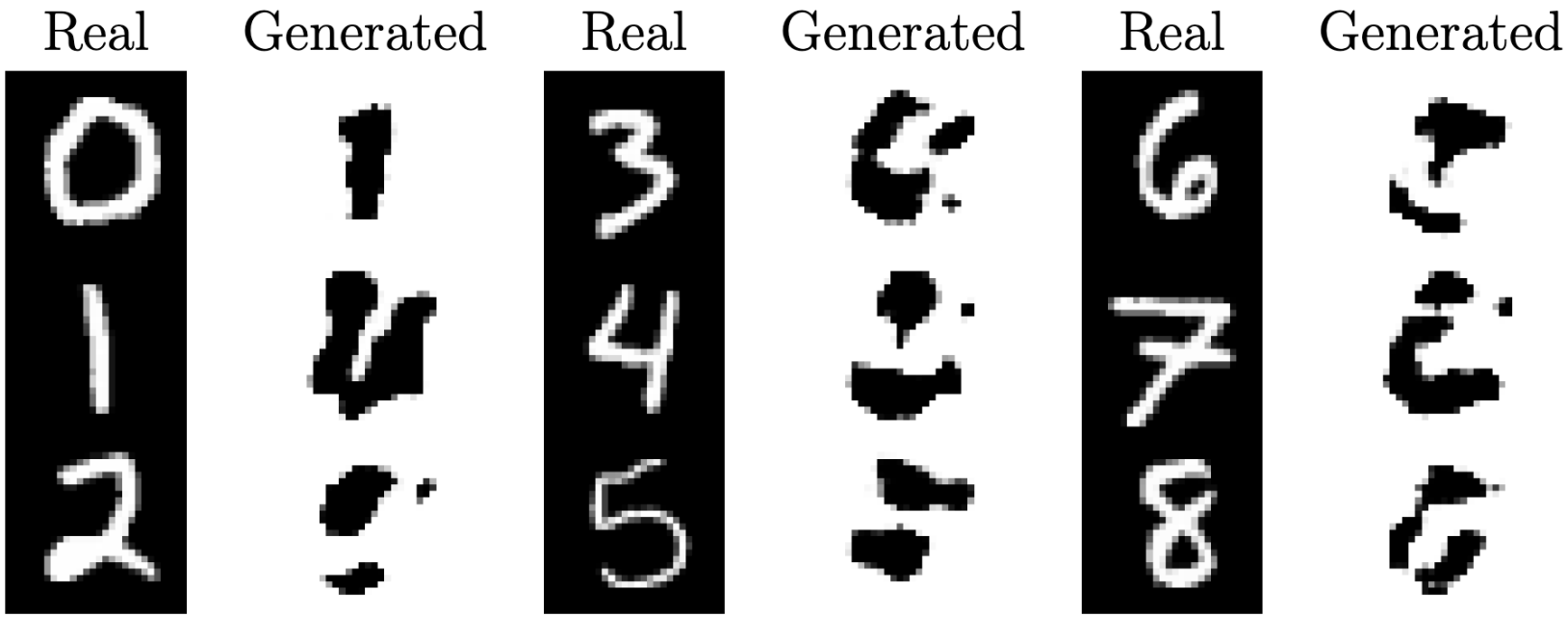


Рис.: Ілюстрація роботи генератора $\mathcal{G}$ на наборі данних *MNIST*.

Приклади зображень. LFW

Рис.: Ілюстрація роботи генератора $\mathcal{G}$ на наборі данних *LFW*.

Генерація криптографічного ключа

Нечіткий екстрактор (Fuzzy Extractor).

Нечіткий екстрактор складається з двох функцій:

- Gen (generate) приймає строку $s \in \{0,1\}^{\ell}$ і видає шифр $r \in \{0,1\}^{L}$ з допоміжною строкою $p \in \{0,1\}^{*}$.
- Rep (reproduce) приймає строку $s' \in \{0,1\}^{\ell}$ та допоміжну строку $p \in \{0,1\}^{*}$, видає шифр $r' \in \{0,1\}^{L}$.

Коректність.

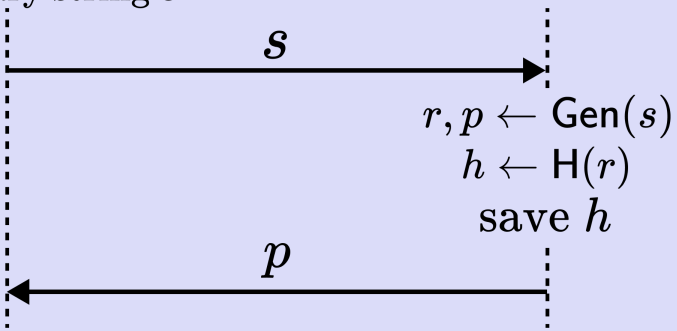
$$\mathbb{P}\begin{bmatrix} r, p \leftarrow \mathrm{Gen}(s) \\ \mathrm{dist}(s, s') < t \\ \mathrm{Rep}(s', p) = r \end{bmatrix} = 1 - \mathrm{negl}(\lambda)$$

На практиці достатньо $t \approx \frac{\ell}{4}$.

Нечіткий екстрактор для системи автентифікації.

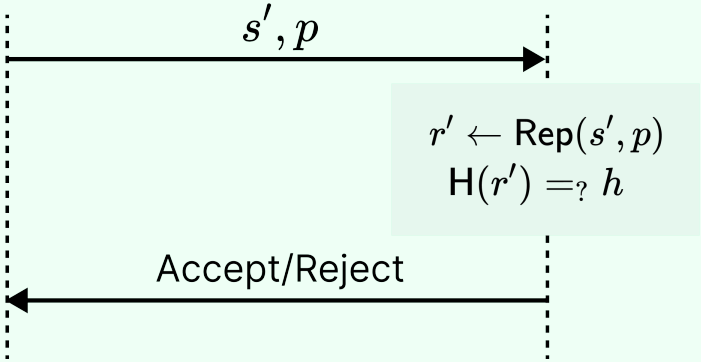
Registration:

Form binary string s



Login:

Form binary string s'



Бінарний екстрактор.

Задача нейронної мережі побудувати функцію $\phi : \text{Image} \to \{0,1\}^\ell$.

Ідея #1

Спочатку знайдемо ембедінги за допомогою FaceNet: $\mathcal{F} : \mathcal{I} \to \mathbb{R}^\ell$, а далі побудуємо $\mathbb{R}^\ell \mapsto \{0,1\}^\ell$.

Ідея #2

Візьмемо знак $\mathcal{F}_i(X), i \in [\ell]$ для отримання бінарної строки: $\phi(X) = \text{Sign}\,(\mathcal{F}(X) > \mathbf{0})$.

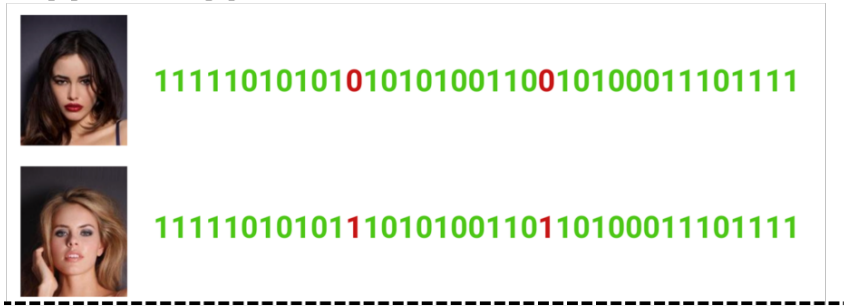
Ідея #3

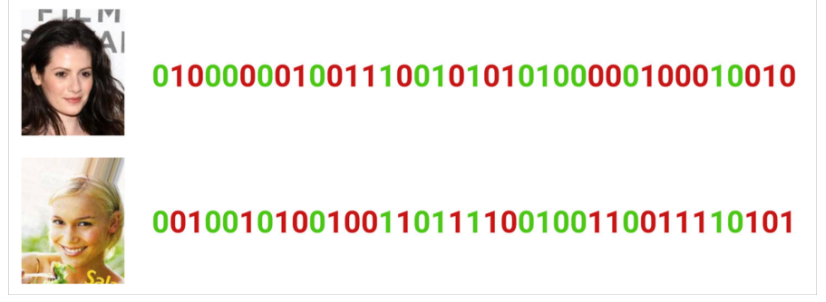
Будемо брати знак відносно $\boldsymbol{\mu} := \mathbb{E}_{X \sim p_{\text{data}}}[\mathcal{F}(X)]$: $\phi(X) = \text{Sign}\,(\mathcal{F}(X) > \boldsymbol{\mu})$.

Результати.

Середня схожість однакових людей – 75%,
Середня схожість різних людей – 50%.

Одні люди





Різні люди

Рис.: Ілюстрація роботи конвертатора у бінарну строку.

Детекція живності

Формулювання

Рис.: Постановка задачі. Спираючись на зображення $X \in \mathcal{I}$, видати ймовірність того, що перед нами нереальна людина.

Архітектура

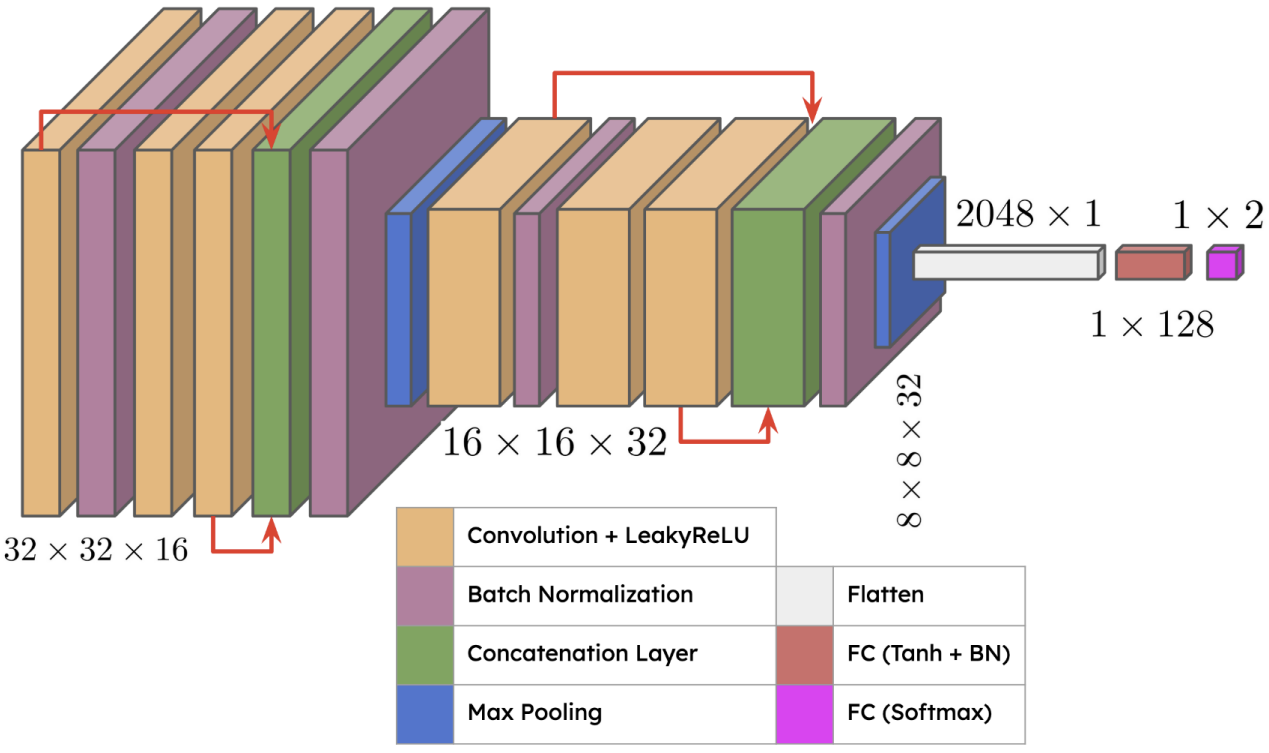


Figure 1: *AttackNet* Architecture [24]

Рис.: Архітектура *AttackNet v2.2*.

Набори данних

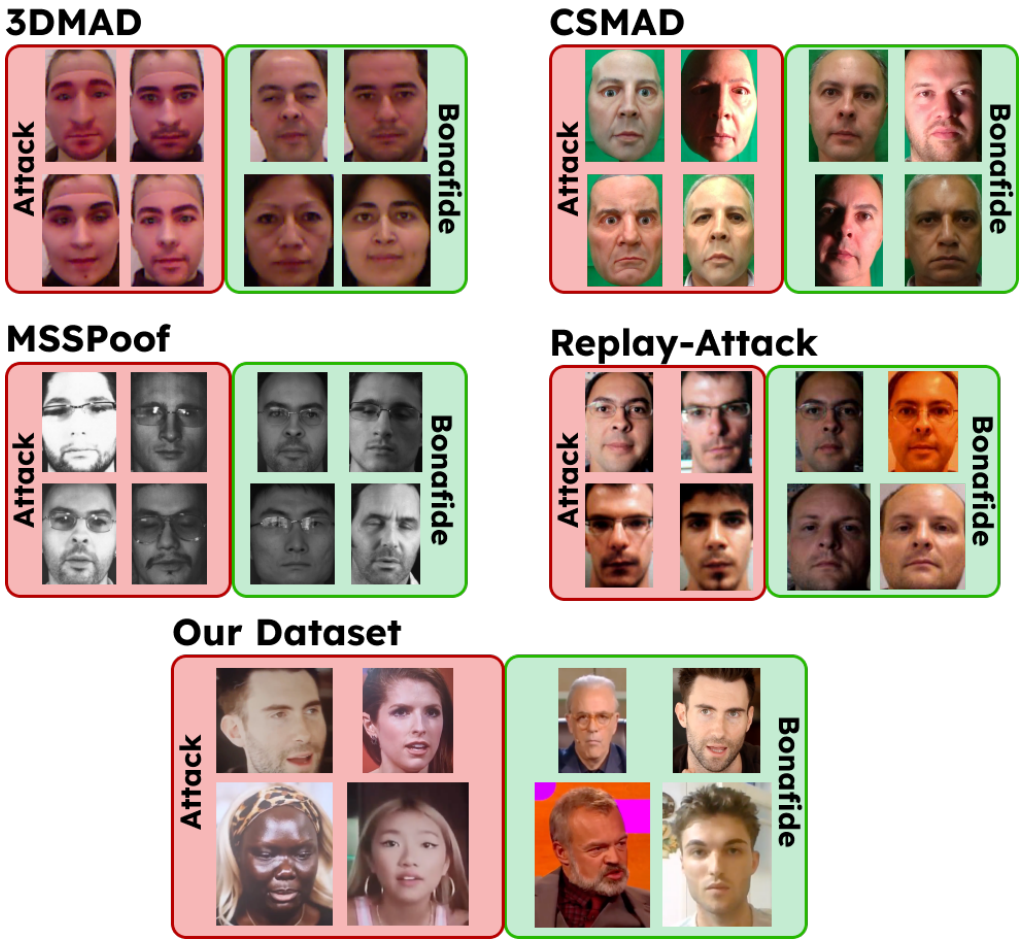


Рис.: Набори данних, на яких проводилось тренування.

Attention Maps

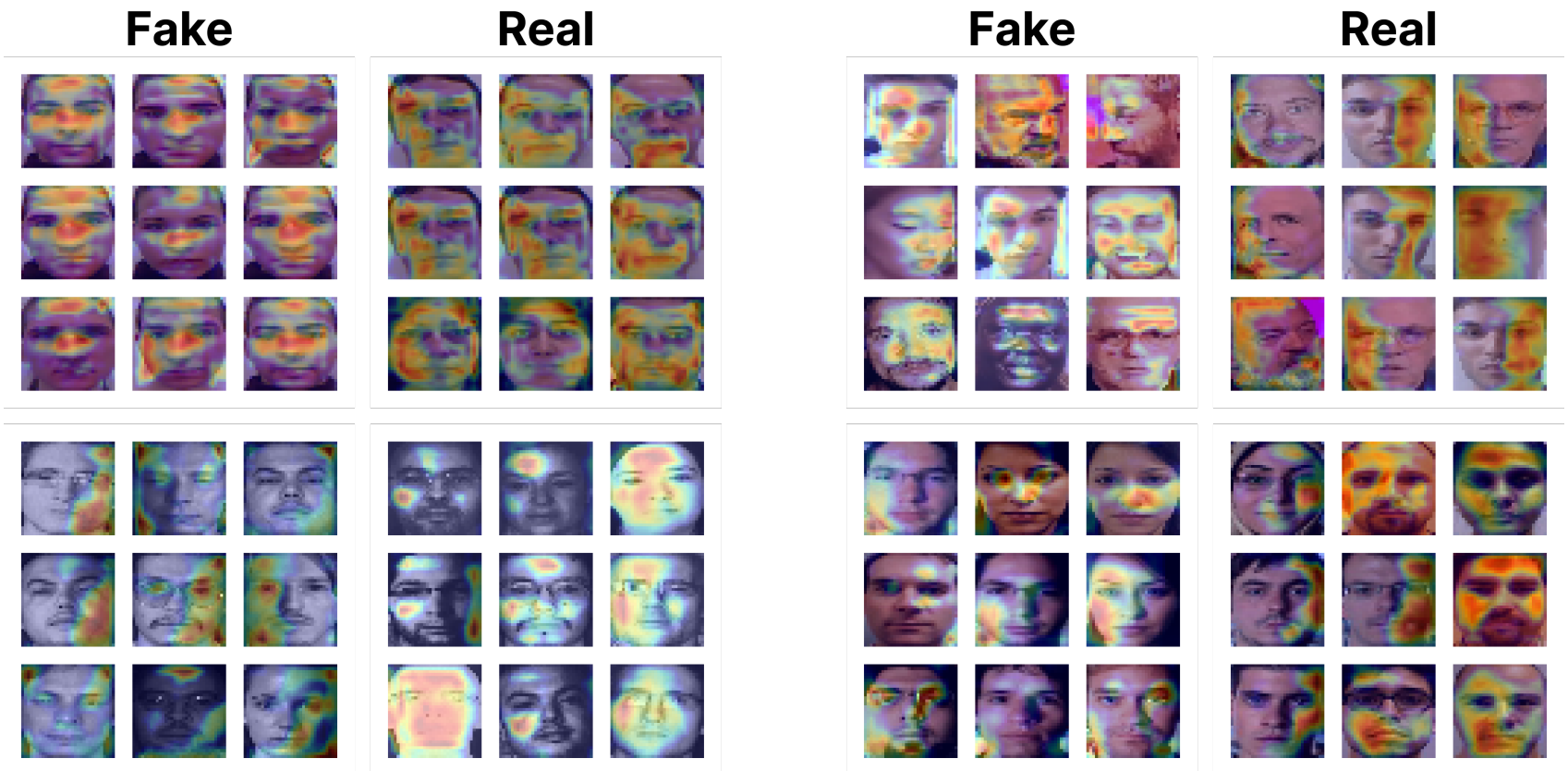


Рис.: Які зони облич нейронна мережа вважає важливими?

Найбільш актуальні напрямки досліджень

- Мультибіометрія.
- Наскільки безпечно використовувати генератор $\mathcal{G}$? Ввести формальне поняття безпеки біометрії в контексті некриптографічно стійких систем.
- Аналог гомоморфного шифрування: чи можна, маючи зашифрований шаблон $\mathcal{G}(X)$, зробити висновки по ньому?
- Перевести обрахунки над $\mathbb{R}$ в операції над скіченними полями $\mathbb{F}_p$: двері до *zero-knowledge proofs* (або zkml).

Дякую за увагу!